Claims

What is claimed is:

1. A method that presents structured entities in a tree-like form based on a set of user provided rules. ~~of representation and manipulation of a structured entity that presents data in a tree-like form based on mapping of every rule from the set of structure defining rules to dynamically or statically created tree-like representation. This mapping is then used for building and manipulating tree-like representation of instances of a structured entity.~~

2. A method according to claim 1 wherein the rules are a BNF/EBNF grammar.

3. A method according to claim 2 wherein the input (instance of a structured entity) is a piece of code in a programming language, or data, or binary file, or a communication protocol message(s).

4. A method according to claim 1 ~~further comprising the steps of: providing a plurality of trees wherein each tree is coupled to its own programmable representation which is based on the structured entity type or content, and changing the representation of structured entities from a conventional state (text) to a related representation state.~~ that builds multiple different tree-like representations for the same structured content based on different sets of mapping rules .

5. A method according to claim 1 further comprising the steps of creating, modifying, ~~manipulating,~~ and deleting mapping rules used for representation building; ~~s. The representations could also be static (Oust based on the rules) or dynamic (build dynamically based on the rules and/or other factors, such as structured entity content). If~~if amapping rules ~~representation~~ changes, the

structured entity representation can be dynamically rebuilt according to the ~~appropriate~~ updated set of rules.~~version of the rule representation.~~

6. A method according to claim 3 further comprising the steps of writing programs or modifying data by ~~creating, modifying, copying, moving, and deleting of single nodes and sets of~~manipulating nodes in ~~such specifically built~~ tree-like representations of ~~the~~ structured entities~~y~~.

7. A method according to claim 3 where representation is built only for some parts of the structured content. ~~A method according to claim 1 further comprising of creating a set of tree-like structures and a way to move pieces of structured entity based on representation to another or the same structured entity to modify it via its tree-like representation.~~

8. A method according to claim 1 further comprising the opportunity to ~~drill-down the nodes of the tree-like representation to~~ create an~~other~~ instance of representation of structured entity that includes only a subset of the nodes of ~~the original~~ a parent ~~structured entity~~representation.
~~. This new instance can be edited separately and dynamically represent the original subset of nodes in the original structured entity or becomes a standalone instance of structured entity, that could be merged back to the original tree later.~~

9. A method according to claim 1 further comprising the opportunity to drill-down the nodes of the tree-like representation besides what the user sees in the original structured entity to expand the underlying properties of the structured entity or expose the related content, which could also be represented according to the claim 1.

10. A method of editing content using tree-like representation according to claim 1 that does not let ~~the~~ user enter the content that does not conform to the related grammar or verifies the changes at the time of the merge with tree-like

representation to determine ~~which ones~~ changes which can ~~and which ones~~or can't be applied, and notifies the user.

11. A method according to claim 1 when the user is not a person, but a program.

12. A method according to claim 1 where instances of a structured entity are modified using both its regular representation and the tree like representation described in the present invention.

13. A method of source control based on the tree-like representation according to the claim 3, when the content is locked for manipulations at the level of nodes or sets of nodes of the representation tree, so that multiple users can work on different parts of the content of the same structured entity ~~(file)~~ simultaneously.

14. A method according to claim 1 where the mapping rules for representations are pre-~~-~~built ~~by~~ for the user according to the anticipated user~~'s~~ needs or preferences.

15. A method according to claim 3 where the tree-like representation for code files is built in such a way that positioning of elements in the tree matches ~~according to~~one of the commonly accepted code indentation rules, so that no matter how the original input is formatted, user can view the representation in a way that is easy to comprehend.

16. A method according to claim 1 where a prerecorded sequence of actions can be applied to a set of nodes of tree-like representation based on the factors common for the set of nodes or their content.